Report on Microservice Design Tools in Context of Domain-Driven Design

Stanislav Zeman

February 2024

Microservice architectures have experienced tremendous growth in the path decade. They are a popular choice for many production systems as they provide many benefits compared to the traditional monolithic architecture. However, designing such systems has no specific agreed-upon methodology and is generally a complex process.

This paper summarizes some of the published approaches and tooling to automate the creation of such designs in the context of Domain-Driven Design. This report is by no means exhaustive.

Bruni Rossi, PhD: FI MUNI

Available at https://github.com/stanislav-zeman/microservice-boundaries

Contents

1. Introduction

2. Domain-Driven Design

Domain-Driven Design (DDD) is a set of practices and an approaches to building and designing software solutions. The DDD helps in aligning the software and it's architecture with bussiness vision, goals and requirements. The DDD as of itself was tossed by Evans (2004).

The DDD as defined coprises of three pillars; ubiquitous language, strategic and tactical design. All of these are further described in the next section.

2.1. Ubiquitous language

2.2. Strategic design

2.3. Tactical design

2.4. Event Storming

3. Automation

3.1. Context Mapper DSL

3.2. Service Cutter

4. Conclusion

References

Evans, Eric. 2004. *Domain-Driven Design: Tackling Complexity in the Heart of Software.*: Addison-Wesley.

List of Figures